

Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Algoritmy pro komunikaci periferních obvodů s mikropočítačem

**Peripheral Circuits of Microcomputer and their Communication
Algorithms**

Zadání bakalářské práce

Student: **Martin Uhlík**
Studijní program: B2649 Elektrotechnika
Studijní obor: 2612R003 Aplikovaná elektronika
Téma: **Algoritmy pro komunikaci periferních obvodů s mikropočítačem**
Peripheral Circuits of Microcomputer and their Communication Algorithms

Zásady pro vypracování:

1. Proved'te rozbor funkce vybraných typů periferních a podpůrných obvodů mikropočítače s ohledem na jejich komunikační protokol.
2. Realizujte zapojení mikropočítače s vybranými periferními obvody.
3. Implementujte potřebné algoritmy pro komunikaci těchto obvodů s mikropočítačem a proved'te jejich popis pro další využití.
4. Proved'te ověření funkce jednotlivých obvodů v jednoduchých konkrétních aplikacích.

Seznam doporučené odborné literatury:

Dle pokynů vedoucího závěrečné práce

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Petr Palacký, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry

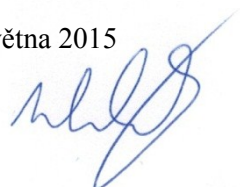


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 4. května 2015

Podpis:

A handwritten signature in blue ink, consisting of stylized, cursive letters, likely representing the author's name.

Abstrakt

Předmětem této práce je rozbor funkce vybraných typů podpůrných a periferních obvodů používaných společně s mikropočítači, přehled některých nejpoužívanějších sériových sběrnic a příklad řešení konkrétních aplikací.

Na vybraných sériových sběrnicích jsou realizovány komunikační algoritmy v jazyce C pro použití na různých platformách mikropočítačů. V práci je uveden příklad jednoduché konstrukce využívající digitální teplotní čidlo.

Hlavním výsledkem této práce je použitelné programové vybavení v jazyce C pro sběrnice I²C, SPI a 1-Wire. Veškeré funkce pro obsluhu sběrnic jsou řešeny softwarově, a proto lze tyto programy po malých úpravách použít na libovolné platformě mikropočítačů.

Klíčová slova

Algoritmy, jazyk c, programování, mikrokontrolér, mikropočítač, sběrnice, interface, periferie, komunikace, sériové sběrnice, SPI, I²C, 1-Wire, Serial Peripheral Interface, Inter-Integrated Circuit.

Abstract

The main object of this thesis is analysis the function of peripheral circuits used together with microcomputers. In this thesis I get overview of the most common serial buses and I get examples solutions in applications.

I realized serial bus communication algorithms in C language for use on different hardware platforms. In this thesis is an example of a simple construction using digital temperature sensor.

The main results of this work are useful software libraries in C language for communication on I²C, SPI and 1-Wire buses. All functions for operating on buses are handled by software implementation for hardware portability. They can be used on any microcomputers platform.

Key words

Algorithms, c language, programming, microcontroller unit, microcomputer, bus, interface, peripherals, communication, serial bus, SPI, I²C, 1-Wire, Serial Peripheral Interface, Inter-Integrated Circuit.

Obsah

1. Úvod	1
2. Rozdělení sběrnic	2
2.1. Paralelní sběrnice.....	2
2.2. Sériové sběrnice.....	2
3. Jednovodičová sběrnice Dallas 1-Wire®	3
3.1. Teplotní senzor DS18B20	4
4. Serial peripheral interface (SPI)	7
4.1. Paměť 25C020.....	8
4.2. A/D převodník MCP3202.....	8
5. Inter-Integrated Circuit (I ² C)	10
5.1. Paměť 24C08.....	11
5.2. Teplotní senzor TC74	13
5.3. Obvod reálného času MCP7940M	13
6. Knihovny pro komunikaci po vybraných sběrnicích.....	15
6.1. Dallas 1-Wire.....	15
6.2. SPI (Serial Peripheral Interface).....	16
6.3. I ² C (Inter-Integrated Circuit)	16
7. Příklad konstrukce s periferním obvodem – venkovní teploměr	18
7.1. Popis zapojení.....	18
7.2. Popis použitého mikrokontroléru	18
7.3. Hlavní body programu.....	20
8. Závěr.....	22
9. Literatura	23
10. Přílohy	24

1. Úvod

Tato bakalářská práce shrnuje možnosti komunikace s periferními obvody mikrokontroléru. Cílem je napsat univerzální knihovny pro obsluhu vybraných sběrnic v jazyce C a odzkoušet jejich funkci.

V první části práce se zabývám popisem nejdůležitějších sběrnic. U každé sběrnice jsou popsány průběhy na vodičích, komunikační protokol a specifika dané sběrnice. Dále je vybrán některý z dostupných periferních obvodů a je stručně popsána jeho funkce. U některých obvodů jsou doplněny i záznamy průběhů z osciloskopu pro ukázkou komunikace.

V další části jsou popsány funkce v napsaných knihovnách pro obsluhu daných sběrnic. Jednotlivé sběrnice mají každá svou knihovnu. Všechny je možno najít v elektronické příloze této práce.

V poslední části je uveden příklad konstrukce venkovního teploměru využívajícího teplotní čidlo DS18B20 a mikrokontrolér PIC16F84A. Program je pro snadnější pochopení napsán v jazyce symbolických adres – assembleru. V této kapitole je stručně popsán i samotný použitý mikrokontrolér.

2. Rozdělení sběrnice

Sběrnice můžeme rozdělit podle vzdálenosti komunikujících zařízení (integrované obvody na jedné desce plošných spojů nebo vzdálená zařízení v průmyslu), podle způsobu přenosu dat (paralelní a sériové), počtem vodičů (jednovodičové, vícevodičové), možnostmi oboustranného přenosu (half-duplex nebo full-duplex), synchronizací hodinovým signálem (synchronní a asynchronní), atd.

2.1. Paralelní sběrnice

Základním znakem těchto sběrnic je přenos více bitů v jednom hodinovém taktu, nejčastěji o délce jednoho slova – 8 bitů. Pro každý bit je tedy nutno vést jeden datový vodič. Kromě datových vodičů se používají také další signály pro řízení toku dat. Z toho tedy vyplývá zásadní charakteristika těchto sběrnic, a to náročnost na počet pinů mikrokontroléru. Tyto sběrnice jsou nevhodné pro použití v systémech, v nichž jsou náročné požadavky na spotřebu a úsporu místa.

Z teoretického hlediska by tyto sběrnice měly mít vyšší přenosovou rychlost, protože se přenáší více bitů současně, v praxi však dochází k rušení a přeslechům mezi datovými vodiči v závislosti na konkrétním provedení sběrnice. Konečná rychlost je tedy menší než u sériových, navíc omezena fyzickou délkou sběrnice.

Jako příklad této sběrnice je možné uvést LPT port v počítačích PC, rozhraní IDE (Paralel ATA) dříve používané pro komunikaci s pevnými disky. Také sběrnice používaná pro ovládání LCD displeje s řadičem HD44780 je paralelní.

2.2. Sériové sběrnice

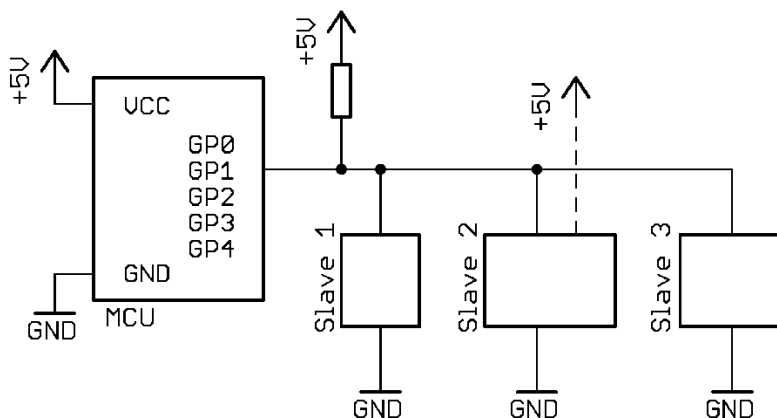
Pro periferní obvody mikrokontrolérů se používají nejčastěji sériové sběrnice, jednak pro úsporu vývodů mikrokontrolérů a také pro jednoduchost řízení. Většina moderních mikrokontrolérů má totiž tyto sběrnice již zabudované jako vnitřní periferie. V programu tedy stačí provést úvodní nastavení po zapnutí a poté se již jen vkládají data do určených registrů. Komunikace následně probíhá automaticky.

Sériové sběrnice můžeme dále dělit podle mnoha různých parametrů, jejichž výčet téměř nikdy není zcela vyčerpávající. Historicky lze poznamenat, že každou sběrnici vyvinula některá z velkých společností zabývajících se vývojem a výrobou polovodičových obvodů pro potřeby komunikace mezi obvody té které firmy.

Nejčastějšími obvody na sériových sběrnících jsou digitální teplotní čidla, externí napětově nezávislé paměti (EEPROM, Flash, ...), rychlé A/D převodníky, apod.

3. Jednovodičová sběrnice Dallas 1-Wire®

Tuto sběrnici uvedla na trh společnost Dallas Semiconductor (dnes Maxim-Dallas). Pro tuto sběrnici je charakteristické, že komunikace probíhá jedním vodičem v obou směrech a připojená zařízení mohou být z této sběrnice napájena. Příklad zapojení sběrnice je na obrázku č. 1. Tato sběrnice může sloužit i pro připojení vzdálených zařízení, [1] uvádí použitelnost do 300 m. S narůstající délkou však roste kapacita vodiče a tím pádem i klesá strmost hran, což může vést k přeslechům. Bezpečnost přenosu dat je proto zajištěná použitím polynomického kontrolního součtu.



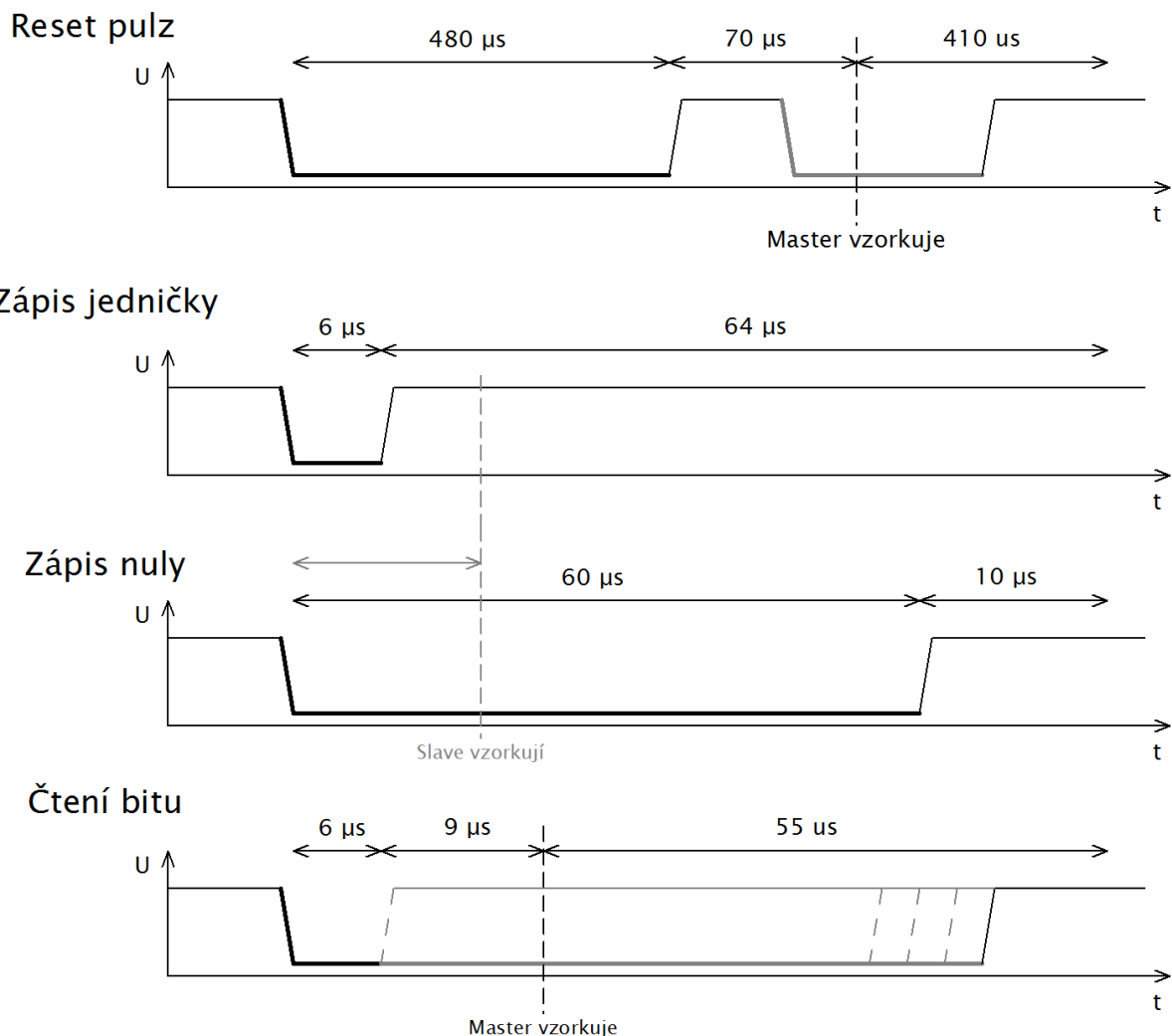
Obrázek 1 – příklad zapojení sběrnice 1-Wire

Na sběrnici je jeden Master (obvykle mikroprocesor) a několik Slave. Každé zařízení Slave má svou jedinečnou adresu, která je tvořena z 8 bitového kódu rodiny (Family code), který určuje, o jaký typ obvodu se jedná a 48 bitového sériového čísla (ROM code), které je pro každý obvod jedinečné. Pro vysoký počet bitů se stejná kombinace bitů sériového čísla prakticky nemůže opakovat po celou dobu výroby čipu.

Komunikaci ovládá vždy Master a zahajuje tak přenos dat jak z, tak i do obvodů Slave. Průběhy napětí na sběrnici při komunikaci jsou na obrázku 2. Zápis i čtení začíná impulzem log. 0 vyslaným mikrokontrolérem po dobu alespoň 5 μ s. Pokud chce mikrokontrolér zapsat nulu, drží linku ve stavu log. 0 i nadále, jinak ji uvolní do log. 1. Při čtení dat mikrokontrolér uvolní linku vždy a po definované době čte stav linky. Jak je tedy z komunikace patrné, je nutné stále mít přehled o tom, ve které části protokolu se komunikace zrovna nachází (jestli bude následovat čtení nebo zápis dat), jelikož tvar signálů pro zahájení zápisu nebo čtení je stejný. Také je nutno mít na paměti, že zařízení Slave jsou většinou napájena ze sběrnice a ponechání linky ve stavu log. 0 (např. z důvodu přerušení mikrokontroléru) způsobí jejich reset.

Komunikace je vždy zahájena tzv. Reset pulzem, na který všechna zařízení Slave zareagují tzv. Presence pulzem a nastaví pozici komunikačního protokolu na začátek. Protokol komunikace je rozdělen na „ROM příkazy“ a „funkční příkazy“. Stručně je možné jej definovat na obrázku 3. Z něj je patrné, že výměna dat probíhá vždy po 8 bitech. Výjimkou jsou pouze dvě funkce pro vyhledání zařízení na sběrnici, které pracují s přenosem po jednom, příp. po dvou bitech.

Toto schéma protokolu nemůže obsahovat výčet všech dostupných funkcí, na něj zde není prostor. Mělo by sloužit spíše k orientaci v protokolu a zdůraznit nutnost Reset pulzu vždy před zahájením jakékoliv činnosti. Podrobnější zobrazení komunikačního protokolu je možné najít jednak v příslušných datasheetech jednotlivých zařízení Slave pro sběrnici 1-Wire, a také v [1] a [2], kde je popis sběrnice 1-Wire vyčerpávající.

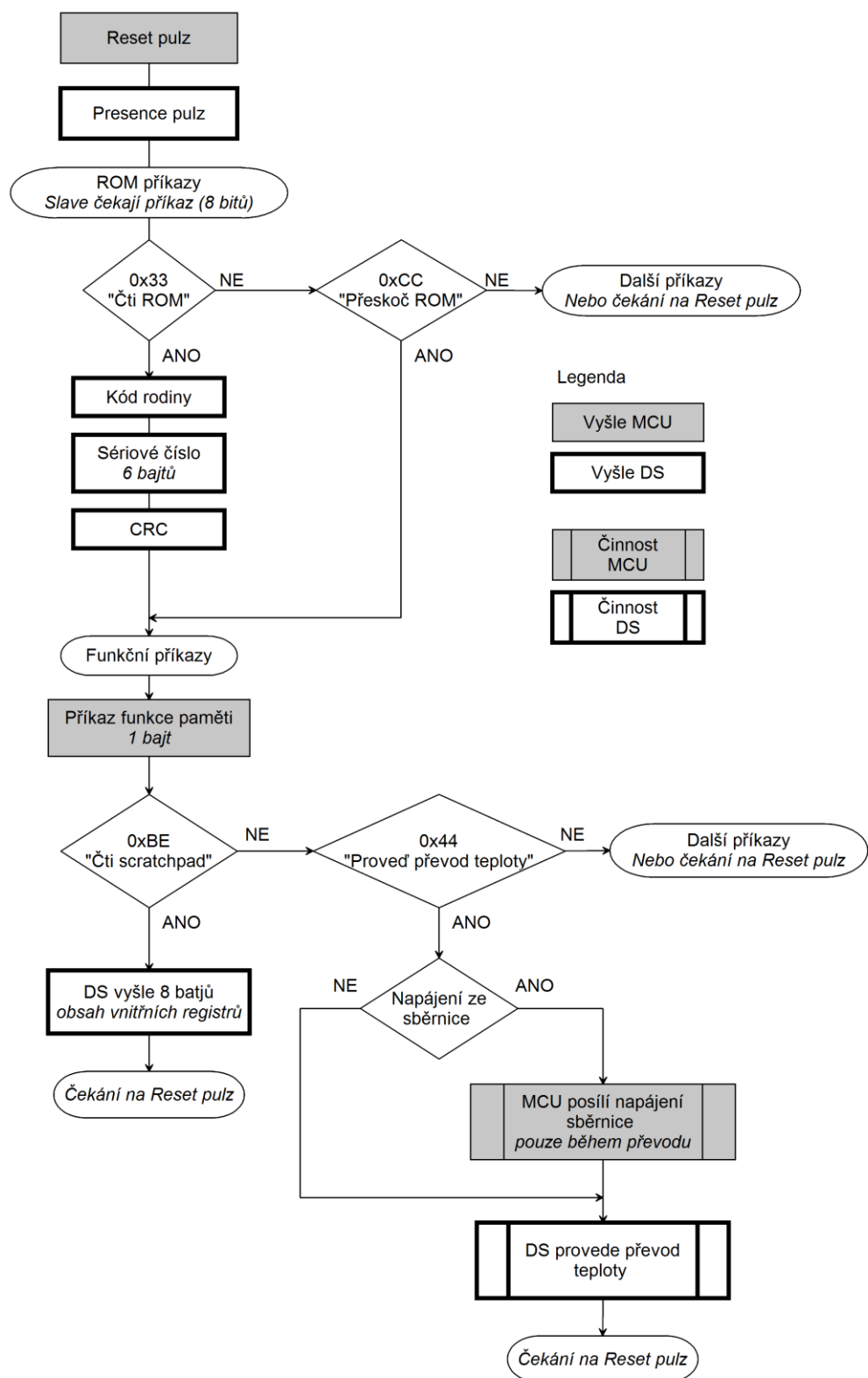


Obrázek 2 – průběhy napětí na sběrnici 1-Wire

Pro tuto sběrnici vyrábí firma Maxim-Dallas celé spektrum obvodů, od již zmíněných přístupových klíčů, přes čítače reálného času, napěťově nezávislé paměti až po teplotní čidla. Nejčastěji využívané obvody jsou právě teplotní čidla. Tato čidla jsou digitální a poskytují výstup přímo jako číslo ve stupních Celsia, není tedy v mikroprocesoru nutné provádět přepočty. Celkově je však komunikace na sběrnici komplikovaná 64bitovým sériovým kódem, který je nutno v mikroprocesoru uložit do paměti (pokud je na sběrnici zařízení více). Časování naznačené na obrázku 2 je závazné, pro několik typů obvodů je však možno použít zrychlený režim, kde jsou povolené kratší časové intervaly průběhů.

3.1. Teplotní senzor DS18B20

Čidlo teploty DS18B20 od výrobce Maxim-Dallas komunikuje po sběrnici 1-Wire. Každý integrovaný obvod má jedinečné sériové číslo, a je tak možno zapojit na sběrnici neomezeně mnoho těchto čidel. Přesnost rozlišení teploty je až 2^{-4}°C ($0,0625^{\circ}\text{C}$). Rozlišení je možné programově zvolit a to v rozsahu 10 až 12bitové. Doba převodu je proměnná podle zvolené přesnosti. Při nejvyšší přesnosti trvá převod maximálně 750 ms, při nejnižší 94 ms. Každý převod teploty je nutno spustit příkazem z mikrokontroléru. Dále obvod umožňuje hlídat rozpětí teplot a případné překročení pak umožní mikrokontrolérem identifikovat.

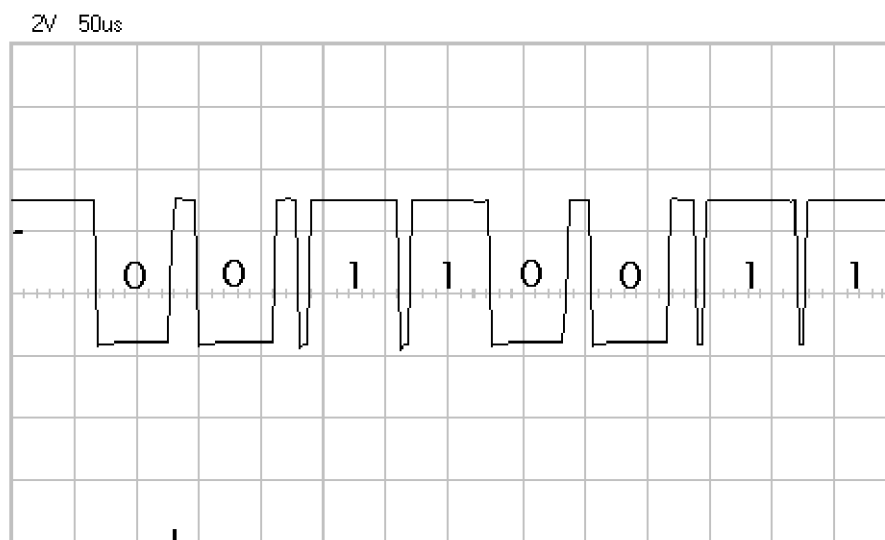


Obrázek 3 – stručný příklad komunikačního protokolu na sběrnici 1-Wire (z [2])

Vývojový diagram komunikačního protokolu je stručně na obrázku 3 nebo kompletní v [3]. Základem jsou ROM příkazy pro identifikaci nebo vyhledání obvodu na sběrnici. V druhé části diagramu jsou pak funkční příkazy pro převod teploty, čtení vnitřních registrů apod.

Obvod DS18B20 obsahuje 8 vnitřních registrů, které výrobce v [3] nazývá scratchpad. V jednotlivých registrech je pak uložena převedená teplota, hystereze alarmu a konfigurace. Tyto registry je možné číst, zapisovat je možné pouze do registrů pro alarm a konfiguraci. Čidlo také disponuje vnitřní pamětí EEPROM, do níž je možno uložit konfiguraci (rozlišení teploty) a hystereze alarmu. Z této EEPROM jsou po zapnutí data automaticky přepsána do scratchpadu.

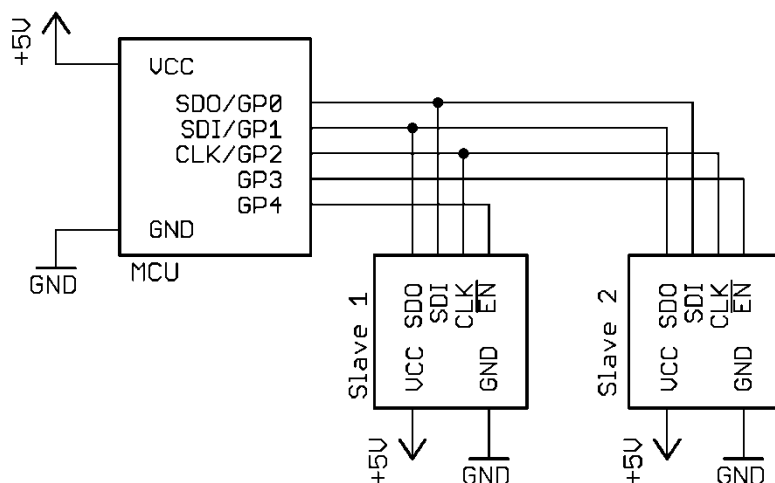
Na obrázku 4 vidíme zápis příkazu 0xCC (přeskoč ROM funkce) do čidla po jednovodičové sběrnici 1-Wire.



Obrázek 4 – přenos bajtu hodnoty 0xCC po sběrnici 1-Wire

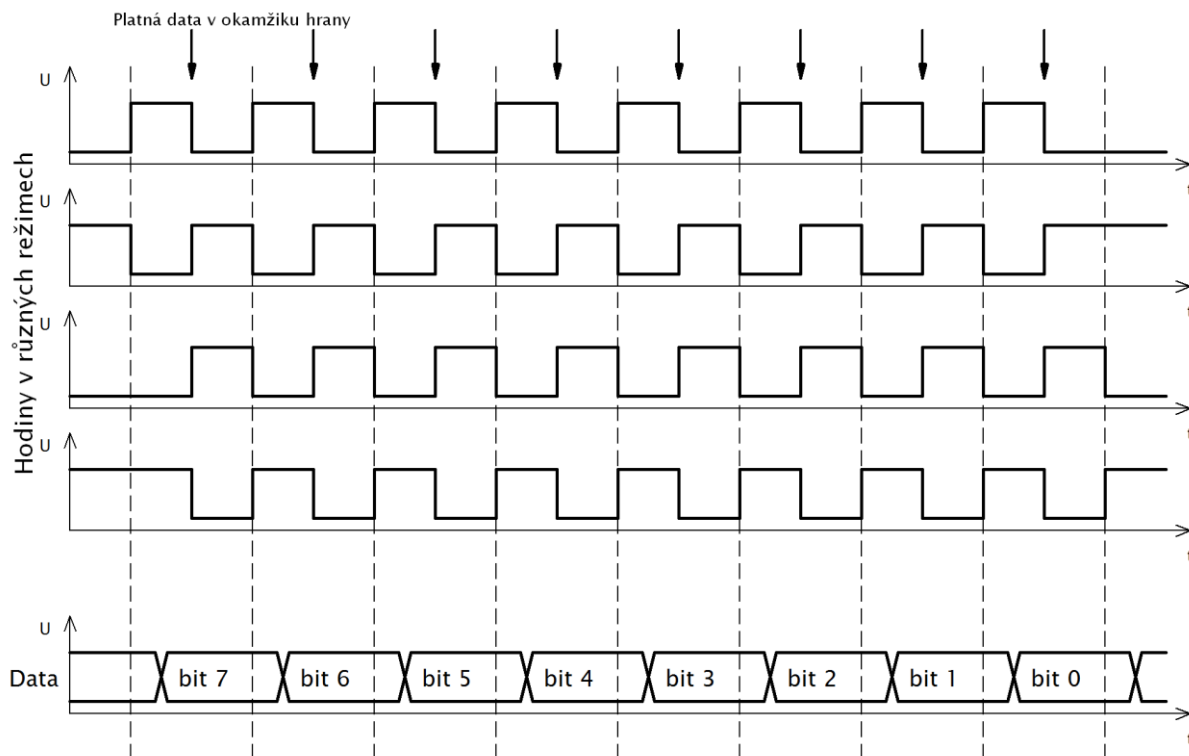
4. Serial peripheral interface (SPI)

Sběrnice SPI je synchronní sériová sběrnice vyvinutá firmou Motorola. Rozhraní s podobným interface také vyvinula firma National Semiconductor pod názvem Microwire. Pro přenos dat používá v každém směru jeden vodič, navíc používá vodič pro aktivaci konkrétního podřízeného obvodu (tzv. Chip Select nebo Enable). Na sběrnici je vždy jeden Master a jedno nebo více zařízení Slave. Princip zapojení vstupů a výstupů je na obrázku 5.



Obrázek 5 – příklad zapojení sběrnice SPI

Co se týče komunikačního protokolu, u sběrnice SPI je na rozdíl od ostatních sběrnic velmi mnoho variant komunikace. Polarita hodinového signálu může být aktivní v log. 0 nebo v log. 1, stejně jako jeho aktivní hrana může být vzestupná nebo sestupná. Samotný přenos je v základu 8bitový, nicméně může nabývat i jiných hodnot (10bitový, 16bitový, atd.). Nejčastěji probíhá od MSB po LSB. Ukázka všech režimů hodinového signálu společně s přenosem dat se nachází na obrázku 6.



Obrázek 6 – varianty přenosu SPI

Díky benevolentnímu protokolu lze na sběrnici SPI připojit libovolný posuvný registr. Může tedy sloužit jako pouhý expandér výstupů mikrokontroléru. Pro tuto sběrnici se rovněž vyrábí velké množství obvodů. Nejčastěji to jsou externí paměti EEPROM, ale také A/D převodníky, teplotní čidla atd. Maximální dosažitelná rychlost závisí na konkrétních typech připojených obvodů. Pohybuje se obvykle kolem 1 MHz, může však nabývat i vyšších hodnot. Nevýhodou této sběrnice je vyšší množství potřebných vodičů pro komunikaci navíc rostoucí s počtem připojených zařízení, kdy každé zařízení má svůj vstup Enable nebo Chip select.

4.1. Paměť 25C020

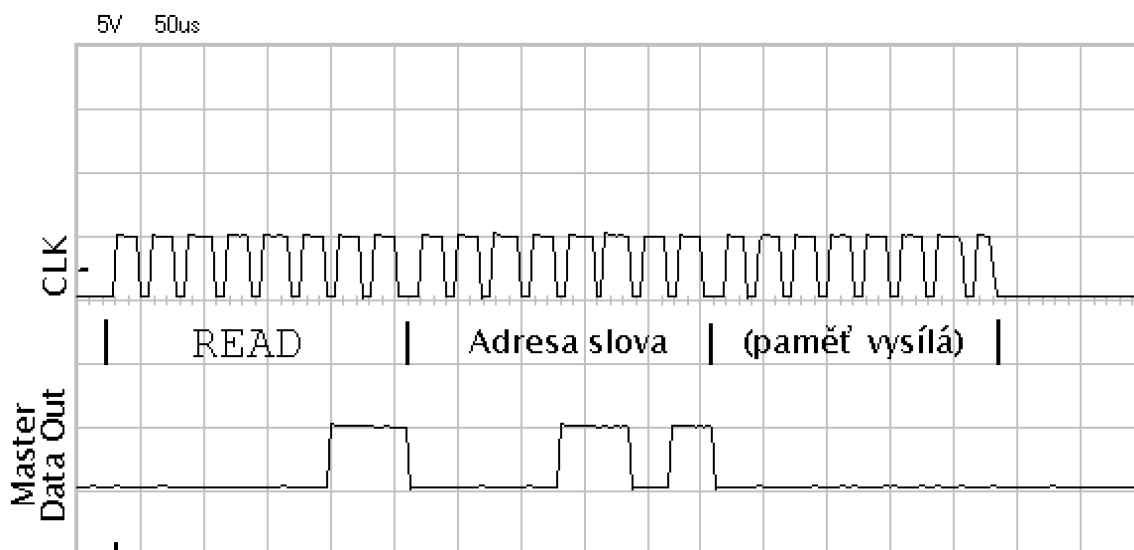
Paměť 25C020 je paměť o velikosti 256 bajtů připojitelná přes sběrnici SPI. Výrobce v [4] uvádí maximální rychlost hodin až 10 MHz, za podmínky napájení napětím 5 V. Disponuje vstupem pro ochranu před zápisem WP (Write Protect) a vstupem pro pozastavení komunikace na sběrnici HOLD. Oba vstupy jsou stejně jako signál CS (Chip Select) aktivní v nule.

Komunikace s pamětí je jednoduchá. Do paměti je nejprve nutno odeslat instrukci podle tabulky 1 převzaté z [4]. Po této instrukci následuje byte s adresou čtené nebo zapisované buňky a poté je již možno odesílat nebo přijímat data. Po instrukcích Write Enable a Write Disable nenásleduje již žádné slovo a je nutno je pouze potvrdit nastavením signálu CS zpět do neaktivní úrovně.

Tabulka 1 – instrukce paměti 25C020 (z [4])

READ	0000 X011	Přečte data (po instrukci následuje adresa)
WRITE	0000 X010	Zapíše data (po instrukci následuje adresa a data)
WRDI	0000 X100	Zakáže zápis
WREN	0000 X110	Povolí zápis
RDSR	0000 X101	Přečte stavový registr (následuje ihned čtení registru)
WRSR	0000 X001	Zapíše stavový registr (následuje ihned zápis registru)

Zápis je zakázán vždy po zapnutí a po každém úspěšném zápisu. Při každém zápisu je tedy nutno opakovaně povolit zápis. Tento postup slouží jako ochrana dat před náhodným přepisem způsobeným např. rušením na sběrnici. Ukázka průběhu čtení z adresy 0x0D je na následujícím obrázku.



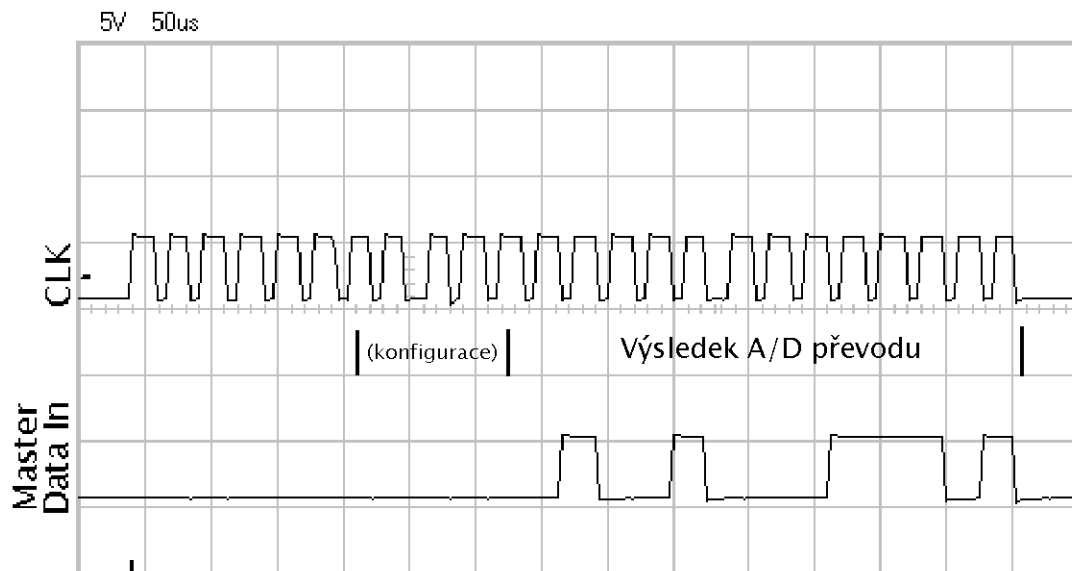
Obrázek 7 – průběh dat při čtení z paměti 25C020

4.2. A/D převodník MCP3202

Ačkoliv dnešní moderní mikrokontroléry již jsou standardně vybavovány A/D převodníkem, externí obvody převodníků mají stále svůj význam. Mohou poskytnout lepší přesnost, diferenciální vstupy nebo větší rozlišení. Zmíněný A/D převodník pracuje na sběrnici SPI a řadí se mezi

aproximační převodníky. Rozlišení je 12 bitů a má dva vstupní kanály, které se mohou konfigurovat také jako diferenciální.

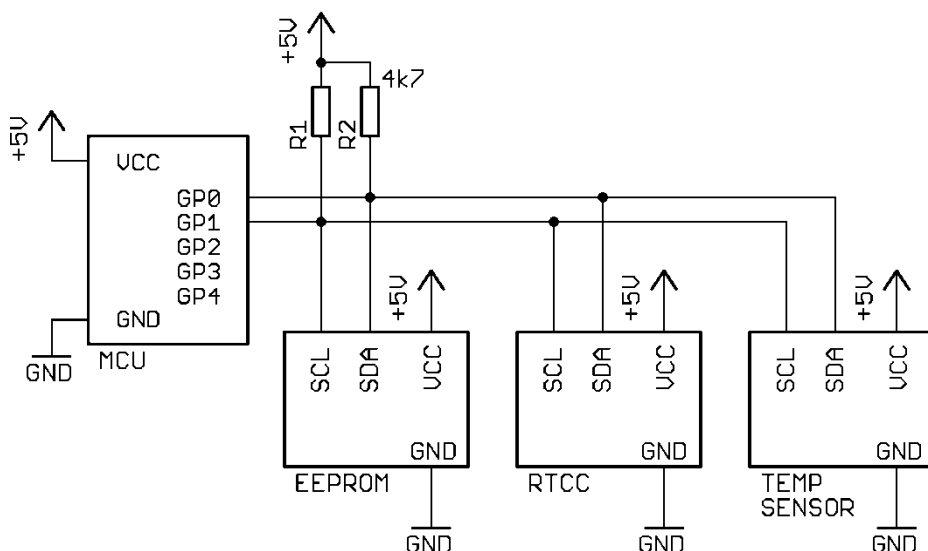
Převod probíhá v reálném čase během rotace dat a dosažitelná hodnota je podle [5] až 100 tisíc vzorků za sekundu při napájení 5 V. Protože časování sběrnice přímo řídí časování převodu, je maximální frekvence na sběrnici omezena na 1,8 MHz při napájení 5 V. Ukázka průběhu dat přijímaných z A/D převodníku je na následujícím obrázku.



Obrázek 8 – průběh dat při čtení z A/D převodníku MCP3202

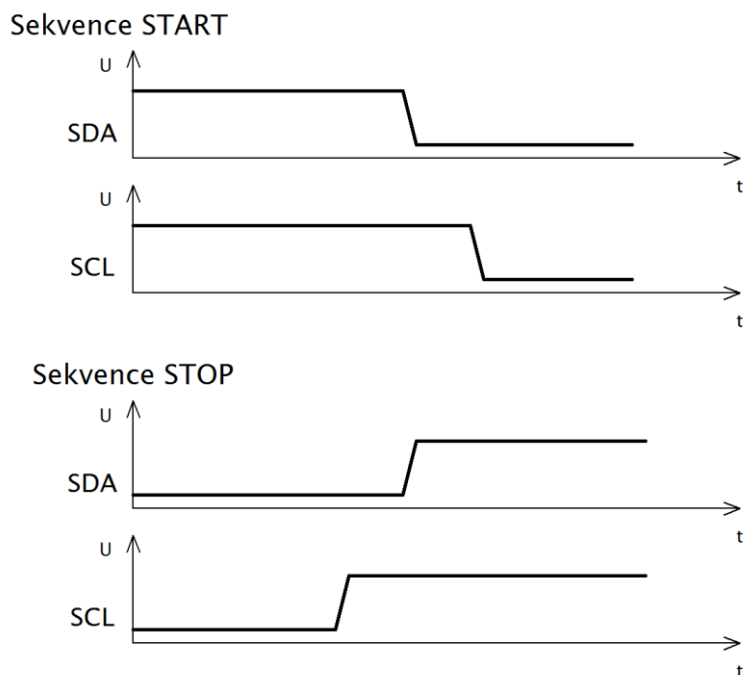
5. Inter-Integrated Circuit (I²C)

Tuto sběrnici vyvinula v 80. letech firma Philips Semiconductors (dnes NXP Semiconductors). Výhodou sběrnice jsou pouze dva vodiče pro přenos oběma směry, synchronní operace pro jednodušší řízení sběrnice a adresný systém pro možnost připojení více obvodů. Na jedné sběrnici může být jeden nebo více zařízení master a několik zařízení slave. Tzv. multi-master režim se však používá jen zřídka a proto se jím nebudu zabývat. Příklad zapojení zařízení na sběrnici je uveden na následujícím obrázku 9.



Obrázek 9 – příklad zapojení sběrnice I²C

Na sběrnici funguje adresný systém a každé zařízení má svou adresu. Ta je nejčastěji 7bitová, existuje však i varianta protokolu s 10bitovou adresou. Ta se však používá jen zřídka. Komunikaci zahajuje i ukončuje master tzv. Start a Stop sekvencí. Tyto sekvence ohraničují datové pakety. Veškerá data se přenášejí vždy po 8 bitech. Po každém bajtu následuje tzv. Acknowledge příznak, aktivní v nule. Tento příznak informuje vysílací stranu, že bajt byl přijímačem řádně přijat a zpracován. Za posledním bajtem má příznak hodnotu 1 (Not Acknowledge) což je signál pro vysílač, že komunikace bude ukončena. Data jsou platná během trvání log. 1 na lince SCL. Pouze během trvání log. 0 na SCL je možné měnit stav linky SDA. Toto pravidlo neplatí pro Start a Stop sekvenci. Ukázky průběhů na obou linkách jsou na obrázku 10.



Obrázek 10 – tvar Start a Stop sekvence na sběrnici I²C

Na začátku každé komunikace master odešle 7bitovou adresu zařízení, s nímž chce komunikovat, a jeden bit určující v jakém režimu chce s obvodem komunikovat. Režimy jsou dva – Read (čtení) a Write (zápis). Pro režim Read je hodnota tohoto bitu 1, pro Write je hodnota 0. V režimu Write master zapisuje data do zařízení slave, v režimu Read je pouze čte. Komunikace v jednom paketu je tedy vždy jednosměrná, zahájená adresou slave a informací o směru toku dat. Pro změnu směru toku dat během paketu je možno vyslat opakovanou sekvenci Start a po ní běžným způsobem adresovat zařízení slave. Ukázka přenosu dat je na následujícím obrázku.

Start	7bitová adresa	R/W	Ack	Data 8 bitů	Ack	Data 8 bitů	NoAck	Stop
-------	----------------	-----	-----	-------------	-----	-------------	-------	------

Obrázek 11 – protokol sběrnice I²C

Adresa zařízení je v obvyklých případech pevná, např. u obvodů reálného času nebo různých čidel, v některých případech ji však lze změnit. Paměti EEPROM pro sběrnici I²C často mají vyvedené piny, jimiž lze adresu paměti ovlivnit. Obvykle tyto piny jsou tři, takže na sběrnici je pak možno připojit až 8 pamětí stejného typu a rozšířit tak jejich kapacitu. Rychlost sběrnice je pevná, ve dvou režimech. Standardní režim 100 kHz a rychlý režim 400 kHz.

5.1. Paměť 24C08

Jak již bylo zmíněno, paměť komunikuje po sběrnici I²C. Její kapacita je 1 kB (8 Kbitů). Protože 1 kB je 1024 bajtů, adresování je nutno řešit 10 bitovým číslem. Proto je prostor rozdělen na čtyři bloky, v každém je 256 bajtů. Adresa paměti na sběrnici I²C je binárně 10100XX, kde poslední dva bity XX označují nejvyšší dva bity adresy slova. Z toho je patrné, že na sběrnici může být připojena nejvýše jedna tato paměť, protože její adresa je pevná.

FIGURE 7-2: RANDOM READ

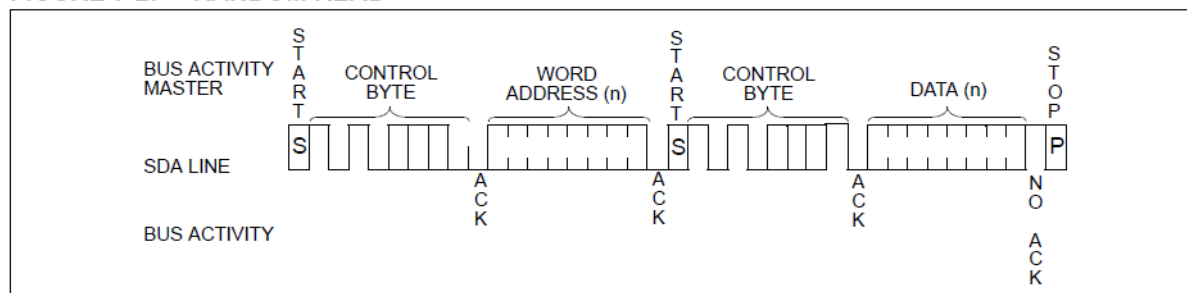
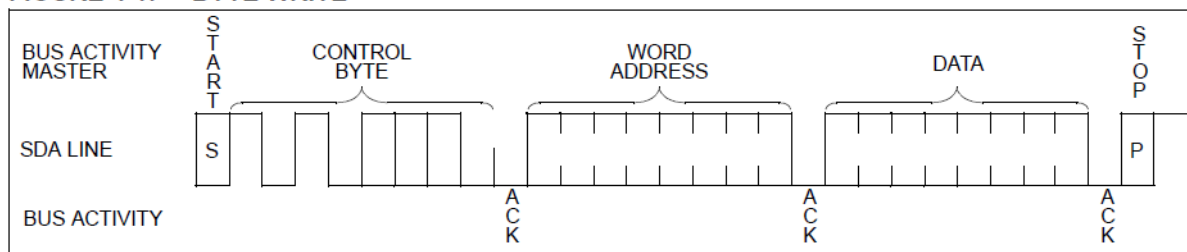


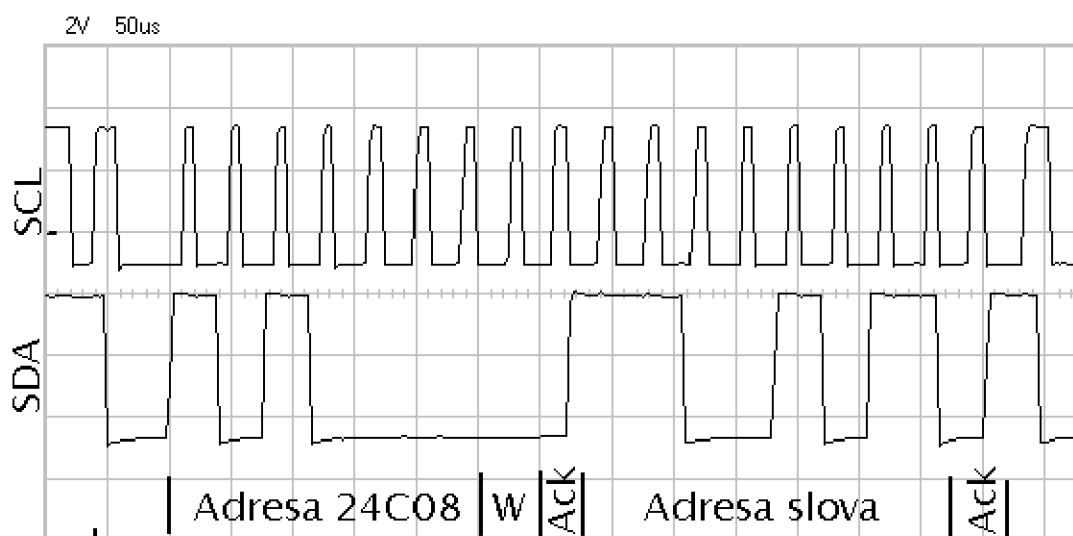
FIGURE 4-1: BYTE WRITE



Obrázek 12 – princip komunikace s pamětí 24C08 (z [6])

Komunikace s pamětí při zápisu a čtení jednoho bajtu je patrná na obrázku 12 převzatého z [6]. Pro zápis se na sběrnici odešle kontrolní slovo (control byte) obsahující adresu obvodu, nejvyšší dva bity slova a bit R/W (v tomto případě v hodnotě 0 – režim Write), dále se odešle zbývajících 8 bitů adresy slova a po něm již mohou následovat data. Na každý bajt paměť odpoví potvrzovacím bitem Acknowledge. Podle [6] je také možno do paměti zapsat až 16 bajtů najednou. V tomto případě však musí jít o slova ve stejné stránce paměti, tedy s posledními čtyřmi bity adresy od 0000 do 1111.

Pro čtení existuje více možností, na obrázku 12 je naznačeno čtení z požadované adresy označované také jako náhodné čtení (random read). Nejprve se do paměti odešle adresa požadovaného slova, zde je postup stejný jako pro zápis. Po odeslání adresy je nutné komunikaci restartovat opětovným vysláním Start sekvenční a následně je již možné přejít do režimu Read a číst data. Na obrázku je naznačeno čtení pouze jednoho bajtu, čtení však může pokračovat libovolně dlouho. Po každém bajtu musí mikrokontrolér odpovědět vysláním bitu Acknowledge, za posledním bajtem však musí odeslat Not Acknowledge. Po nepotvrzení příjmu se paměť odpojí od sběrnice a mikrokontrolér může ukončit komunikaci sekvencí Stop.



Obrázek 13 – průběhy na sběrnici I²C při komunikaci s 24C08

Při testování knihoven pro komunikaci jsem zaznamenal na osciloskopu průběh napětí na linkách SDA a SCL při adresaci paměti. Je možno jej spatřit na obrázku 13. Na tomto obrázku je naznačena adresa paměti a adresa slova. Ze zobrazených průběhů je patrné, že adresa slova byla 0x0CB (v hexadecimální podobě).

5.2. Teplotní senzor TC74

Jednoduchý teplotní senzor komunikující po sběrnici I²C. V [9] výrobce uvádí rozlišení je 1°C a rychlost převodu je typicky 8 vzorků za sekundu. Převody teploty probíhají nepřetržitě od okamžiku zapnutí obvodu. Na rozdíl od typu DS18B20 popsaného v předchozí kapitole není nutné odesílat příkaz pro provedení převodu teploty.

Adresa obvodu na sběrnici je 1001xxx, kde poslední tři bity určuje číslo v názvu konkrétního obvodu. Obvod TC74A0 má tyto bity rovny 000, obvod TC74A7 má tyto bity 111. Je tedy možné zakoupit čidla až s osmi různými adresami, z čehož vyplývá možnost připojení až osmi čidel na jednu sběrnici.

Adresování vnitřních registrů a přístup k nim je obdobný jako u již zmiňované sériové paměti. Adresa registru s teplotou je 0x00, adresa konfiguračního registru je 0x01. Pro čtení z konkrétního registru je nutné napřed v režimu Write nastavit adresu požadovaného registru, restartovat komunikaci a v režimu Read je možné opakovaně z tohoto registru číst.

V konfiguračním registru se nachází pouze dva významové bity. Bit určující režim operace (zastavení převodů a snížení spotřeby) a bit pouze ke čtení, který indikuje platná data (je nastaven na jedničku ihned po prvním převodu teploty).

Formát teploty je se znaménkem – dvojkový doplněk. Podobně jako u čidla DS18B20. Nejvyšší měřitelná teplota je +127 °C, nejnižší pak -65 °C.

5.3. Obvod reálného času MCP7940M

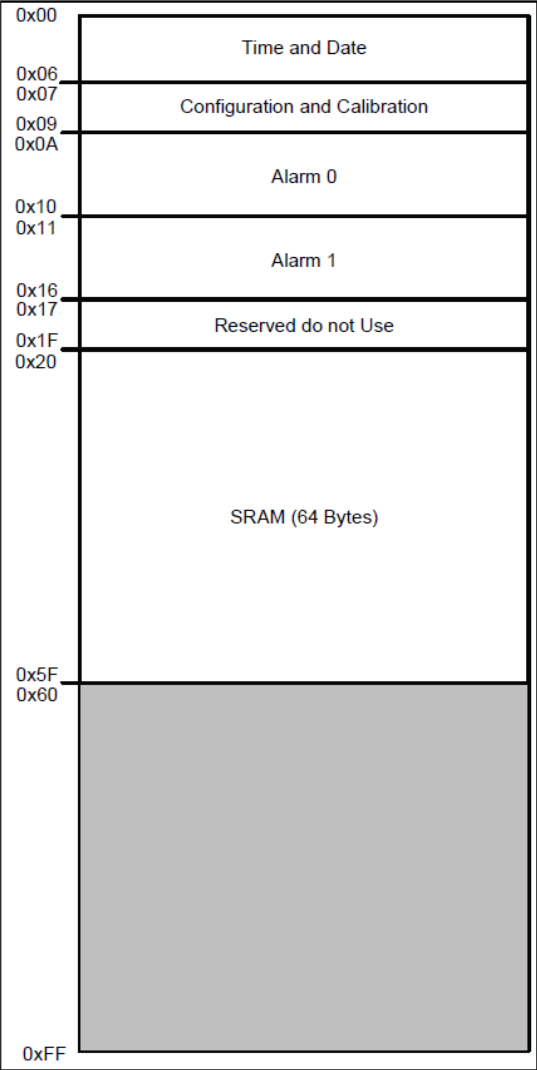
Tzv. obvod RTC (real time counter) je určen na zajištění časové informace v běžně používaných jednotkách – hodiny, minuty, dny, měsíce a roky. Často jsou tyto obvody vybaveny obvody pro napájení z baterie, případně i její nabíjení. Právě napájení z baterie je důvodem snahy výrobců o minimalizaci spotřeby těchto obvodů.

Obvod MCP7940M má kromě samotné časovací funkce také k dispozici dva alarmy, které je možné vyvést na zvláštní pin. Ten je pak možné připojit např. na vstup externího přerušování mikrokontroléru a významně tak snížit jeho spotřebu, kdy mikrokontrolér čeká na přerušování v režimu nízké spotřeby. Dále obvod MCP7940 obsahuje 64 bajtů statické paměti RAM k volnému použití (např. uložení nastavení programu apod.). V klidovém stavu, kdy na sběrnici neprobíhá komunikace a v obvodu je aktivní pouze oscilátor, je jeho spotřeba pouhých 1 µA.

Přístup k jednotlivým registrům a k paměti SRAM je totožný jako u přechodících obvodů na sběrnici I²C. Na obrázku 14 je mapa paměti převzatá z [7]. Na začátku paměťového prostoru jsou umístěny registry obsahující časovou informaci. Tyto informace jsou uloženy v BCD formě.

Po základních registrech s časem jsou v paměti registry pro nastavení a kalibraci oscilátoru. Dále pak registry pro první a druhý alarm a nakonec 64 bajtů paměti k volnému použití. Jelikož se jedná o paměť SRAM, přístupové doby čtení a zápisu jsou velice nízké, stejně tak i spotřeba energie.

FIGURE 4-1: MEMORY MAP



Obrázek 14 – mapa paměti obvodu MCP7940M (z [7])

6. Knihovny pro komunikaci po vybraných sběrnicích

Všechny knihovny jsou napsané v jazyce C a byly vyzkoušeny na platformě mikroprocesorů PIC od firmy Microchip. Definice jsou upraveny pro použití v prostředí MPLAB IDE a kompilátoru XC8, je však velmi jednoduché provést potřebné úpravy pro jiná prostředí a jiné hardwarové platformy.

V souborech knihovny *.h jsou umístěny definice pinů pro jednotlivé sběrnice a definice všech funkcí se základním stručným komentářem pro rychlé vyhledání významu každé funkce. V souborech *.c jsou pak deklarace jednotlivých funkcí a u každé z nich je podrobný popis vstupních parametrů a návratové hodnoty funkce. Ve většině funkcí jsou také provedeny komentáře uvnitř pro snadnou orientaci a pochopení funkce.

6.1. Dallas 1-Wire

Knihovnu pro komunikaci na jednovodičové sběrnici Dallas jsem převzal z oficiální stránky výrobce Maxim Integrated (www.maximintegrated.com), kde ji uvádí pod názvem Application Note 187. Podstatnou částí této aplikační poznámky je vyhledávací algoritmus zařízení na jednovodičové sběrnici. Jelikož je tento algoritmus poměrně komplikovaný, nebudu jej zde uvádět a odkážu zájemce na kvalitní popis v českém jazyce v knize [2].

Součástí vyhledávacího algoritmu jsou však také nezbytné funkce pro přenos bitů a bajtů. Celý okomentovaný program lze najít v souborech OneWire.h a OneWire.c umístěnými v elektronické příloze této práce. Zde se zaměřím na popis nejdůležitějších funkcí.

OWReset()

Tato funkce vyšle Reset pulz pro nastavení komunikačního protokolu na začátek vývojového diagramu. Po tomto pulzu vzorkuje sběrnici a očekává Presence pulz, což je odpověď zařízení Slave na úspěšný Reset. Funkce vrátí hodnotu 0x01, pokud se na sběrnici Presence pulz objevil, tedy pokud je na sběrnici alespoň jedno aktivní zařízení.

OWWriteByte()

Funkce vyšle 8 bitů na sběrnici v pořadí od LSB po MSB. Slovo pro odeslání se předává v parametru funkce. Funkce nemá návratovou hodnotu.

OWReadByte()

Funkce přijme 8 bitů ze sběrnice v pořadí od LSB po MSB. Funkce nemá parametry. Návratová hodnota je přijatý bajt.

crcCalc()

Tato funkce počítá polynomický CRC. V parametru funkce se předá právě přijatý bajt, ze kterého je nutné CRC vypočítat. Funkce z tohoto bajtu vypočítá kontrolní součet a uloží jej jednak do globální proměnné crc8 a jednak jej vrátí jako návratovou hodnotu. Na konci přenosu musí být hodnota vypočteného CRC rovna nule. V opačném případě došlo při přenosu k chybě a data jsou neplatná.

OWFirst(), OWNext()

Tyto funkce slouží k vyhledávání zařízení s neznámým sériovým číslem. Po provedení funkce se nachází právě nalezené číslo v poli ROM_NO[]. Toto sériové číslo je možné uložit (celé nebo jen část) v mikrokontroléru a poté je již možné obvody rychle identifikovat.

OWSearchKnown()

Funkce očekává část adresy v poli ROM_NO[]. Po provedení funkce je celé pole ROM_NO[] vyplněné sériovým číslem identifikovaného obvodu a protokol je v místě, kde identifikované zařízení vyčkává na funkční příkaz. Vzhledem k počtu kombinací a malé pravděpodobnosti opakování stačí 3 až 4 bajty. Přičemž z toho je jeden bajt kód rodiny, pevný pro všechny obvody jednoho typu. V paměti mikrokontroléru tak lze použitím této funkce ušetřit mnoho místa ukládáním jen části sériových čísel.

DSTeplota()

Ukázka provedení převodu teploty a přenosu registrů do mikrokontroléru. Funkce předpokládá na sběrnici jediné teplotní čidlo DS18B20. Odešle do něj příkaz k provedení převodu teploty. Po stanovené době převodu vyšle příkaz pro přečtení vnitřních registrů (tzv. scratchpadu). Přečte všechny registry, přičemž průběžně počítá CRC. Přijatou teplotu přepočte na celé číslo v rozsahu -55 až +125 °C. Pokud došlo k chybě při přenosu a vypočtené CRC nebude na konci nulové, funkce vrátí hodnotu +85 °C.

6.2. SPI (Serial Peripheral Interface)

Knihovna pro softwarové ovládání sběrnice SPI se nachází v souborech `spi_sw.h` a `spi_sw.c`. Ukázka funkcí pro čtení a zápis do paměti EEPROM typu 24C020 a funkcí pro A/D převod s převodníkem typu MCP3202 se nacházejí v souborech `spi_prikklady.h` a `spi_prikklady.c`. Všechny tyto soubory lze opět nalézt v elektronické příloze této práce.

Na sběrnici SPI může z principu sběrnice (samostatný vodič pro každý směr) probíhat komunikace oběma směry současně. Proto je funkce pro vysílání a příjem dat společná. V souboru `spi_sw.h` je nutné pomocí definic `CLK_IDLE` a `CLK_ACT` nastavit žádanou polaritu hodinového signálu v klidu a druh aktivní hrany (sestupnou nebo vzestupnou).

spi_init()

Tato funkce pouze nastaví klidovou polaritu hodinového signálu a vynuluje datový výstup. Volá se pouze po zapnutí mikrokontroléru a po nastavení portů.

spi_data()

Univerzální funkce pro přenos dat po sběrnici. Odešle a přijme data od MSB po LSB. Odesílaný bajt se předává funkci jako parametr, přijímaný bajt funkce vrátí jako návratovou hodnotu.

ReadEE()

Přečte bajt z paměti EEPROM typu 24C020. Adresa se předává jako parametr, přečtený bajt funkce vrátí jako návratovou hodnotu.

WriteEE()

Tato funkce zapíše jeden bajt do paměti 24C020. Adresa i bajt k zápisu se předávají jako parametry funkce. Funkce nemá návratovou hodnotu.

getADC()

Funkce provede převod napětí ze zvoleného kanálu A/D převodníku typu MCP3202. Číslo kanálu se předává jako parametr, může nabývat hodnot 0 nebo 1. Z tohoto také vyplývá, že A/D převodník pracuje v režimu jednoho vstupu (nikoliv tedy v diferenciálním režimu). Návratová hodnota je číslo od 0 do 4095, kde číslo 4095 značí plné napájecí napětí A/D převodníku.

6.3. I²C (Inter-Integrated Circuit)

Pro knihovnu ovládání sběrnice I²C platí podobné názvy jako u předchozí knihovny. V souborech `i2c.h` a `i2c.c` se nacházejí funkce pro ovládání sběrnice, v souborech `i2c_prikklady.h` a `i2c_prikklady.c` se nacházejí ukázky práce s pamětí 24C08 a teplotním čidlem TC74A0. Soubory je opět možné nalézt v elektronické příloze práce.

Pro převod na příslušnou hardwarovou platformu je nutné upravit funkce `i2c_high_sda()`, `i2c_low_sda()`, `i2c_high_scl()` a `i2c_low_scl()`. Dále je nutno upravit definice pinů určených pro připojení sběrnice `SDA_PIN` a `SCL_PIN`.

i2c_start(), i2c_stop()

Tyto funkce slouží pro zahájení a ukončení komunikace předepsanými sekvencemi Start a Stop. Funkci `i2c_start()` je možno použít rovněž pro opakovaný start v případě změny směru komunikace (ze zápisu na čtení).

i2c_in()

Funkce přijme jeden bajt ze sběrnice a předá jej jako návratovou hodnotu. Po vykonání této funkce je nutné provést funkci i2c_nack() nebo i2c_ack() a informovat tak vysílač o potvrzení příjmu bitem Acknowledge.

i2c_out()

Funkce vyšle jeden bajt, který je nutno předat jako parametr. Funkce vrací hodnotu bitu Acknowledge. Je-li tato hodnota rovna nule, zařízení potvrdilo příjem. V opačném případě nedošlo k potvrzení příjmu.

EEwrite()

Funkce slouží pro zápis do paměti typu 24C08. V parametrech je nutné předat 10bitovou adresu slova a zapisovaná data.

ERead()

Funkce vrátí data z paměti 24C08. Adresu čteného slova je nutno předat v parametru funkce.

TempRead()

Tato funkce slouží pro obsluhu teplotního čidla TC74A0. Jako návratovou hodnotu předá teplotu ve znaménkovém formátu, v rozsahu -40 až +125 °C. Funkce předpokládá, že čidlo není v režimu Stand-by, ale v režimu normální operace po zapnutí.

7. Příklad konstrukce s periferním obvodem – venkovní teploměr

Jako ukázkou práce mikrokontroléru s periferním obvodem jsem vybral jednoduchý venkovní teploměr s mikrokontrolérem PIC16F84A a teplotním senzorem DS18B20. Tento teploměr je navržen s dobře čitelným 7segmentovým LED displejem a bateriovým napájením. Po zapnutí napájení tlačítkem se provede převod teploty a získané číslo se převede na displej. Na displeji je číslo zobrazené až do odpojení napájení uvolněním tlačítka.

7.1. Popis zapojení

Schéma zapojení teploměru můžeme vidět na obrázku 1 v příloze č. 1. Obsahuje minimum prvků díky použití mikrokontroléru a digitálního teplotního senzoru.

Displej je ovládán multiplexně. Mikrokontrolér PIC16F84A z portu B spíná přímo katody sedmisegmentových displejů přes ochranné rezistory. Anody displejů jsou ovládány přes tranzistory T1 až T4. Tranzistory jsou MOSFET s indukovaným P kanálem. Desetinná tečka u druhého místa na displeji DISP2 je pevně připojená na zem přes ochranný rezistor.

Mikrokontrolér je řízen krystalem o frekvenci 4 MHz. Digitální teplotní čidlo DS18B20 je připojeno pomocí sběrnice 1-Wire k mikrokontroléru. Využívá režimu externího napájení, kdy čidlo není napájené ze sběrnice, ale z vnějšího zdroje. Celé zapojení je napájeno z destičkové baterie 9 V, a proto je na vstupu použit nízkoubytkový stabilizátor LF50. Zapínání a vypínání probíhá tlačítkem umístěným mimo desku plošného spoje a proto není na schématu.

Deska plošného spoje pro teploměr je navržena do elektroinstalační krabice na omítku a její klišé i s osazením je na obrázcích 2 a 3 v příloze č. 1. Čidlo je připojeno třížilovým kabelem vedeným pod omítkou kolem okenního rámu a čidlo je vně budovy zataveno do smršťovací bužírky. Finální podoba výrobku je na obrázku 4 v příloze č. 1.

7.2. Popis použitého mikrokontroléru

Použitý mikrokontrolér PIC16F84A je z rodiny 8bitových mikrokontrolérů firmy Microchip. Je založen na architektuře RISC. Vnitřní blokové schéma převzaté z [8] je zobrazeno na obrázku 15 na následující stránce.

7.2.1. Jádru

Na schématu vidíme základní součásti mikrokontroléru a jeho vnitřní periferie. Jak je z obrázku patrné, mikrokontrolér obsahuje programovou paměť o velikosti 1024 instrukcí, datovou paměť EEPROM o velikosti 64 bajtů a RAM paměť o velikosti 68 bajtů. Nezbytnou součástí mikrokontroléru jsou také vstupně výstupní porty, aritmeticko-logická jednotka (ALU) a zásobník o velikosti 8 skoků, díky němuž lze program rozdělit na podprogramy volané instrukcí CALL.

Protože se jedná o jádro RISC, všechny datové operace v ALU jsou prováděny s tzv. Work registrem (dále budu označovat jako W registr). Pomocí tohoto registru je také možné předávat data mezi podprogramem a hlavním programem.

Veškeré periferie jako porty, časovač atd. se ovládají pomocí speciálních funkčních registrů (SFR) namapovaných do paměťového prostoru společného s RAM. Obsluhují se tedy všemi instrukcemi, které umožňují pracovat s RAM. Jelikož zde není prostor na podrobný výčet instrukcí a jejich funkcí, mohu odkázat jednak na [8] nebo další vhodnou literaturu.

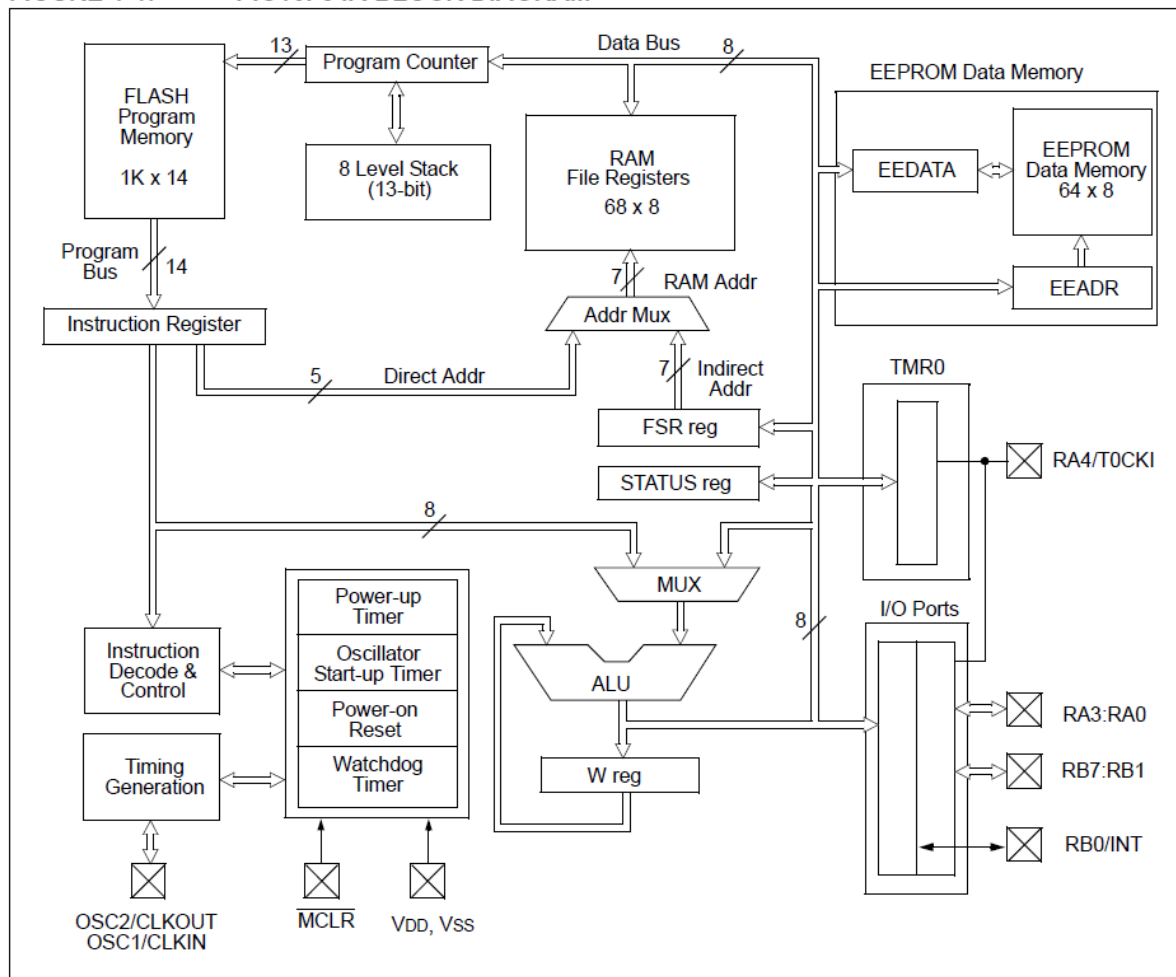
Mikrokontrolér také obsahuje systém přerušení. Zdrojem pro aktivaci přerušení může být změna stavu na vstupu vyhrazených pinů portu B, časovač, ukončení zápisu do vnitřní paměti EEPROM a další. Vektor přerušení je společný na adrese 0x04. Systém přerušení nemá rozlišení priority a rozdělení přerušení podle zdroje je nutné vyřešit programově.

7.2.2. Oscilátor a časování

Časování mikrokontroléru zajišťuje oscilátor nebo zdroj externího hodinového signálu. Oscilátor může být s krystalem nebo s RC článkem. Vzhledem k dnešním nízkým cenám krystalových výbrusů je vhodnější použít vždy krystal. Zajišťuje lepší stabilitu kmitočtu a umožňuje daleko vyšší taktovací frekvence než použití RC článku. Procesor obsahuje tzv. pipelining, což je ve stručnosti

přečtení následující instrukce z paměti programu v okamžiku vykonávání předchozí instrukce. Díky tomu proběhne vykonání každé instrukce ve čtyřech taktech oscilátoru. Rychlost provádění instrukcí je tedy čtvrtina frekvence oscilátoru. Tento kmitočet ($F_{osc}/4$) se poté používá pro periférie jako např. časovač.

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



Obrázek 15 – vnitřní blokové schéma mikrokontroléru PIC16F84A (z [8])

7.2.3. Porty

Mikrokontrolér PIC16F84A má dva vstupně/výstupní porty, port A a B. Port A je 5bitový, port B je 8bitový. Některé piny portů jsou sdíleny se speciálními funkcemi vnitřních periférií, např. externí přerušení, vstup časovače, apod. Tyto funkce se nastavují v SFR jednotlivých periférií. Samotná orientace portů je určena registry s názvem TRISA a TRISB. Bity v těchto registrech přímo určují, které piny budou nastaveny jako výstup a které jako vstup. Samotné ovládání výstupů a čtení vstupů probíhá skrze registry PORTA a PORTB. Je možné s nimi pracovat jako s jakoukoliv pamětovou buňkou a provádět logické operace.

7.2.4. Časovač/čítač

Mikrokontrolér obsahuje také modul časovače čítače nazvaný Timer0. Jedná se o 8bitový registr. Může být nastaven do několika režimů. Vstupem může být vnitřní hodinový signál nebo externí signál přivedený na pin 4 portu A (dále budu piny označovat způsobem RA4 – port A, bit 4). Dále může být na vstupu předřazena předdělička. Ta může nabývat dělicích poměrů 1:2, 1:4, 1:8, ... až 1:256. Čítač může po přetečení z hodnoty 0xFF do 0x00 vygenerovat přerušení. Dá se tedy vhodným nastavením dosáhnout pravidelného přerušení. Tento režim je použit i v konstrukci teploměru. Nastavení čítače Timer0 se provádí pomocí registru OPTION_REG.

7.3. Hlavní body programu

Program je pro snadné pochopení funkce mikrokontroléru napsán v jazyce symbolických adres Assembler. Celý výpis programu je v elektronické příloze této práce. Program je bohatě okomentovaný, proto zde popíšu hlavní části programu s odkazem na jeho celý výpis.

Po resetu začíná mikrokontrolér vykonávat instrukce od adresy nula. Protože program používá přerušení, které má vektor 0x04, je nutné z adresy 0x00 přeskočit obsluhu přerušení na začátek hlavního programu. Tam se nachází inicializace mikrokontroléru a poté samotná komunikace s čidlem.

7.3.1. Počáteční nastavení

Do portů A a B zapíše 0xFF. Všechny připojené součástky jsou aktivní a po nastavení orientace pinů registry TRISA a TRISB se tranzistory nesepnou a displej zůstane zhasnutý. Pomocí registru INTCON nastaví systém přerušení, ale ještě přerušení zcela nepovolí. Pomocí registru OPTION_REG nastaví časovač Timer0 do režimu čítání hodinového signálu z oscilátoru ($F_{osc}/4$) s předděličkou o poměru 1:8. Při taktu krystalu 4 MHz je tedy časovač čítán frekvencí 125 kHz. Frekvence přetečení časovače z 0xFF na 0x00 je přibližně 488 Hz.

7.3.2. Komunikace s DS18B20

Po inicializaci následuje ihned komunikace s čidlem DS18B20 po sběrnici 1-Wire. Protože je čidlo na sběrnici pouze jedno, nepracuje se se sériovým číslem a tato část protokolu se přeskakuje příkazem „Přeskoč ROM“. Do čidla se napřed odešle konfigurace – požadavek na 12bitovou přesnost převodu. Poté je odeslán příkaz k provedení převodu teploty. Díky externímu napájení z čidla může mikrokontrolér kontrolovat stav převodu čtením bitu z čidla. Běží-li převod, čidlo odesílá hodnotu 0, po dokončení odesílá 1. Po dokončení převodu následuje poslední fáze komunikace s čidlem, a to přenos dat do paměti mikrokontroléru.

7.3.3. Vyhodnocení přijatých dat

Hned po příjmu všech bajtů ze scratchpadu program vyhodnotí správnost vypočteného kontrolního součtu v registru CRC, který musí mít nulovou hodnotu. Pokud došlo při přenosu k chybě, registr CRC nebude nulový a program proto skočí na adresu CHYBA. Pokud je vše v pořádku, přistoupí program k převodu dat z 12bitového binárního formátu na dekadický formát v kódu BCD.

Pokud je teplota záporná, převede číslo ve dvojkovém doplňku zpět na řádné binární číslo. Po vyhodnocení znaménka následuje oddělení čtyř bitů desetinného místa od celočíselné části. Převod desetinného místa probíhá pomocí předdefinované tabulky. Celočíselnou část program dělí opakovaným odčítáním nejprve stovky a pak desítky. Zbytek po dělení jsou jednotky. Všechny takto získaná čísla postupně program ukládá na displej. Protože je 7segmentový displej připojen přímo k mikrokontroléru, je nutné před odesláním převést BCD číslo na kód odpovídající rozložení vývodu a segmentů tak, aby se na nich zobrazilo požadované číslo. Tento převod se také provádí pomocí převodní tabulky.

Po provedení všech těchto převodů program povolí přerušení a zůstane v tzv. věčné smyčce, kde nevykonává žádnou činnost.

7.3.4. Zobrazení dat na displej – přerušení

Displej je obsluhován v rámci přerušení, které periodicky vyvolává časovač Timer0. V přerušení je načten vždy příslušný bajt s obsahem v 7segmentovém formátu a odeslán na port. Poté je příslušný displej aktivován sepnutím tranzistoru v anodě. Po této operaci přerušení končí a vrátí se zpět do hlavní smyčky, kde je však program ukončen věčnou smyčkou. Procesor tedy čeká na další přetečení časovače Timer0.

V přerušení je vždy pouze přepnut segment na další, všechny segmenty se postupně rozsvítí během 4 period přerušení. Přerušení je voláno s frekvencí přibližně 488 Hz, tedy každé 2 ms. Jeden segment tedy svítí po dobu 2 ms. Všechny segmenty se postupně rozsvítí za dobu 8 ms, což je frekvence 125 Hz. Tento kmitočet je dostatečně vysoký pro setrvačnost oka a nelze tedy rozeznat, že je displej multiplexovaný.

7.3.5. Podprogram CHYBA

V případě, že dojde k chybě při přenosu dat a je vypočítáno nenulové CRC nebo nedojde k přijetí Presence pulzu, je zavolán podprogram chyba. Podprogram uloží do registrů pro displej kódy písmen Err a podle druhu chyby (CRC nebo Presence) nastaví před displejem znaménko na – nebo +. Nakonec povolí obsluhu přerušení a ukončí činnost věčnou smyčkou. Na displeji se tedy zobrazí kód +Err v případě chyby výpočtu CRC nebo –Err v případě nepřijetí Presence pulzu.

8. Závěr

V této práci jsem v první části popsal tři nejvýznamnější sběrnice používané pro komunikaci mikrokontroléru s periferními obvody. Jsou to Dallas 1-Wire, SPI a I²C. U každé sběrnice je uveden komunikační protokol a jednotlivá specifika. Dále jsou také uvedeny příklady periferních obvodů pro každou sběrnici i s ukázkou funkce.

Ve druhé části práce jsou realizovány knihovny pro ovládání popsaných sběrnic pomocí mikrokontroléru. Knihovny jsou napsány v programovacím jazyce C. Pro vybrané periferní obvody jsou také realizovány knihovny pro ukázkou obsluhy funkcí těchto obvodů. Jsou to funkce pro čtení a zápis do sériových pamětí, funkce pro převod teploty do paměti mikrokontroléru apod.

V poslední části jsem se zabýval příkladem konkrétní konstrukce využívající periferní obvod komunikující po sériové sběrnici. Je popsán venkovní teploměr s mikrokontrolérem PIC16F84A využívající čidlo DS18B20. Schéma zapojení je jednoduché díky použitému digitálnímu čidlu. Program je pro snazší pochopení napsaný v jazyce symbolických adres Assembler a jsou popsány důležité body tohoto programu.

Funkce vytvořených knihoven byla experimentálně ověřována v testovacích zapojeních. Knihovny jsou napsané univerzálně a lze je jednoduše s malými úpravami použít pro libovolnou hardwarovou platformu. Do budoucna je možné tyto knihovny rozšířit např. o podporu funkcí dalších periferních obvodů. Také je možné vytvořit obsluhu sériových portů zabudovaných přímo v mikrokontroléru.

9. Literatura

- [1] – Hrbáček, Jiří. Komunikace mikrokontroléru s okolím – 1. díl. 2. dotisk 1. vydání. Praha: BEN – technická literatura, 2002. ISBN 80-86056-42-2
- [2] – Hrbáček, Jiří. Komunikace mikrokontroléru s okolím – 2. díl. 1. vydání. Praha: BEN – technická literatura, 2000. ISBN 80-86056-73-2
- [3] – DS18B20 Programmable Resolution 1-Wire Digital Thermometer [online]. USA: Maxim Integrated Products, 2008 [citováno 2. května 2015]. Dostupné z: <datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [4] – 25AA020A/25LC020A 2K SPI Bus Serial EEPROM [online]. USA: Microchip Technology Inc., 2003-2012. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21833G.pdf>>
- [5] – MCP3202 2.7V Dual Channel 12-Bit A/D Converter with SPI Serial Interface [online]. USA: Microchip Technology Inc., 1999-2011. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21034F.pdf>>
- [6] – 24C08B/16B 8K/16K 5.0V I²CTM Serial EEPROMs [online]. USA: Microchip Technology Inc., 2004. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21081G.pdf>>
- [7] – MCP7940M Low-Cost I²CTM Real-Time Clock/Calendar with SRAM [online]. USA: Microchip Technology Inc., 2012-2014. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/20002292B.pdf>>
- [8] – PIC16F84A 18-pin Enhanced FLASH/EEPROM 8-Bit Microcontroller [online]. USA: Microchip Technology Inc., 2001-2013. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/35007C.pdf>>
- [9] – TC74 Tiny Serial Digital Thermal Sensor [online]. USA: Microchip Technology Inc., 2001-2012. [citováno 2. května 2015]. Dostupné z: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21462D.pdf>>

10. Přílohy

Všechny přílohy jsou k dispozici také v elektronické formě na přiloženém CD.

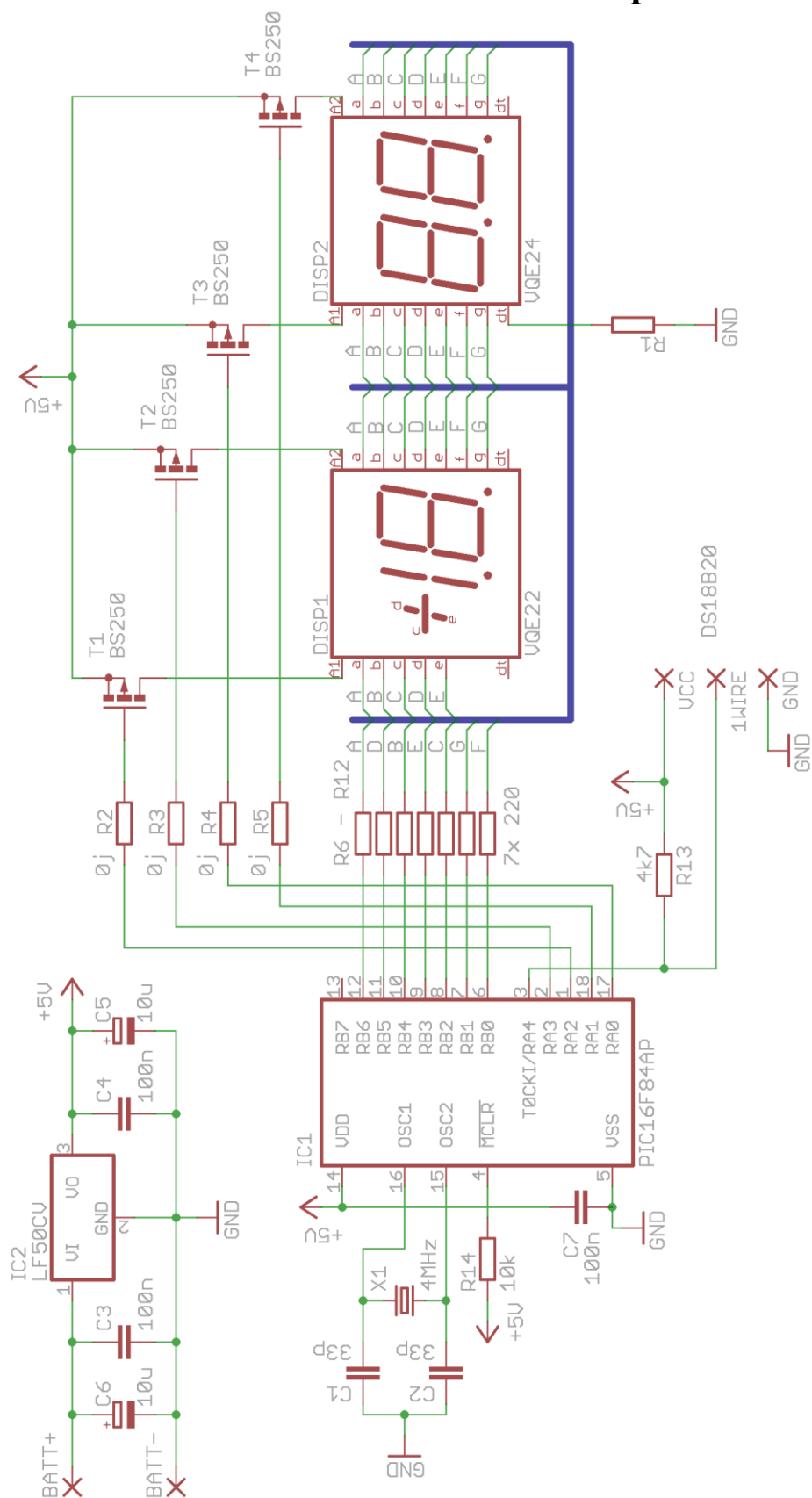
Příloha 1 – schéma zapojení venkovního teploměru, deska plošného spoje a finální provedení

Příloha 2 – výpis programu pro obsluhu venkovního teploměru (*pouze v elektronické podobě*)

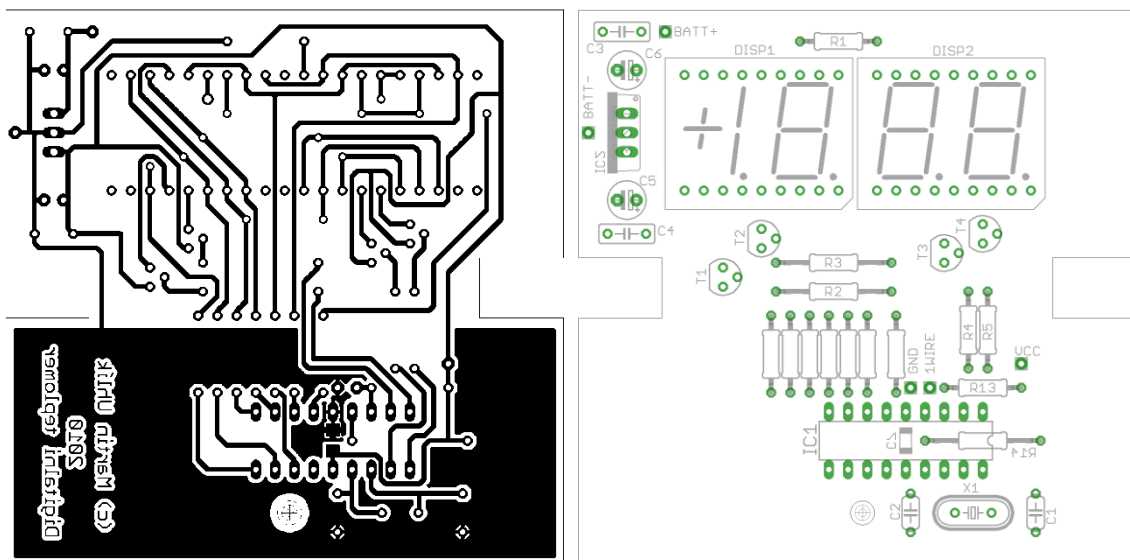
Příloha 3 – knihovny pro komunikaci po sběrnících 1-Wire, SPI a I²C (*pouze v elektronické podobě*)

Příloha 4 – datasheety všech popsanych součástek ve formátu pdf (*pouze v elektronické podobě*)

Příloha 1 – Provedení venkovního teploměru



Obrázek 1 – celkové schéma zapojení



Obrázek 2 (vlevo) – klíše DPS
Obrázek 3 (vpravo) – osazení DPS součástkami



Obrázek 4 – finální provedení teploměru